PROGRAMMABLE LOGIC DEVICES HAVING ENHANCED CASCADE FUNCTIONS TO PROVIDE INCREASED FLEXIBILITY

PRIORITY CLAIM

[1] This application claims priority from Indian patent application No. 693/Del/2002, filed June 27, 2002, which is incorporated herein by reference.

TECHNICAL FIELD

[2] The invention relates generally to programmable logic devices having enhanced cascade functions to provide increased flexibility.

BACKGROUND

[3] A Programmable Logic Device (PLD) comprises a number of relatively simple logic modules with an arrangement to interconnect them in any of a wide variety of ways through a general purpose interconnection network to perform logic functions which can be quite complex. In addition, some of the logic modules include additional logic elements for concatenating the outputs of multiple modules to perform relatively complex logic functions without having to make use of the general purpose interconnection network. This additional logic is termed "Cascade Logic" and is used to implement high-speed, simple logic functions involving a large number of inputs.

[4] **US Patent 5,258,668** discloses a method for cascading logic units in which each logic module includes additional logic elements for forming a logical combination of the normal output signal of that logic module and the output signal from another, adjacent logic module. The output signal from the other logic module is applied directly to the additional logical element in the first logic module. The output signal of the additional logic elements in each logic module becomes the output signal of the logic module. As shown in **FIG. 1,** Block ***20*** is a 4-input look up table (LUT) providing an output ***32,*** connected to the input of cascade logic element ***22.*** The second input to the cascade logic element ***22*** is the cascade output ***44*** of another, preferably adjacent, programmable logic block (PLB). The cascade logic element ***22*** can be any desired logic gate, such as an AND gate. The logic element ***22*** logically combines the two signals ***32*** and ***44*** and applies the result either to a D

flip flop or to the Cascade input ***44*** of the next PLB. The cascade output ***44*** or the flip flop output ***38*** is inverted and can be used as feedback for the LUT ***20***, and also serves as the logic module output ***42.*** This method does not provide the flexibility to use the LUT output ***32*** and the cascade function output ***44*** simultaneously. That is, the LUT output is not available for other logic functions if the cascade function is implemented. In such situations, additional LUT logic is necessary to produce the required output, resulting in increased cost and delay.

[5] **FIG. 2** describes another prior-art implementation in which the cascade input ***72*** is gated by elements ***74a*** and ***74c***. Element ***74b*** is a programmable bit, which is programmed to indicate whether or not connection of the cascade in input to the cascade module ***60*** is desired. If connection of the cascade input ***72*** to the module ***60*** is desired, bit ***74b*** is programmed to enable transistor ***74a*** and disable transistor ***74c***. This applies the cascade input signal ***72*** to logic element ***60***, which here is an **AND** gate. The other input of block ***60*** is the output of LUT ***50***. The AND gate applies the ANDed output of the two inputs to node ***76,*** which is the cascaded output. If connection of the cascade input ***72*** to the cascade logic ***60*** is not required, then the bit ***74b*** is programmed to disable transistor ***74a*** and enable transistor ***74c***. This applies Vcc to the second terminal of the AND gate, thereby allowing that gate to pass the output of the LVT ***50*** to the flip flop ***70***. But as with the arrangement of **FIG. 1**, in this arrangement only one of the outputs, that is, either the cascaded output or the LUT ***50*** output is available at any one time.

Summary

[6] An embodiment of the invention provides an efficient method for cascading that simultaneously provides the cascaded output and normal output of the logic module.

[7] Another embodiment of the invention selectively provides inverted and non-inverted cascade inputs for the cascade logic.

[8] These embodiments include a Programmable Logic Device (PLD) incorporating a plurality of Programmable Logic Blocks (PLBs) providing enhanced flexibility for cascade logic functions, each comprising a multi-input Look Up Tables (LUT) providing one input to a cascade logic block for implementing the desired cascade logic functions. The other input of the cascade logic block is a cascade-in signal. A 2-input selection multiplexer receives one input from the output of the cascade logic block and the other from the output of the LUT for selecting either the cascade logic output or the LUT output as the unregistered output. The cascade logic output and the multiplexer output are simultaneously available from the PLB.

[9] Another embodiment of the invention further includes a flip-flop connected to the output of the selection multiplexer providing registered output to a 2-input output multiplexer for selecting either the unregistered output or the registered output as the final output of the PLB. A feedback arrangement connects the final output to the input of the LUT to enhance the flexibility of the cascade logic as well as the normal functions of the PLB.

[10] Yet another embodiment of the invention includes a 2-input cascade input multiplexer for selecting the cascade-in signal in either its inverted or non-inverted form as one input to the cascade logic.

Brief Description of the Drawings

[11] Features and advantages of the invention will become more apparent in reference to the following description and the accompanying drawings, wherein:

[12] **FIG. 1** shows prior art as disclosed in US patent **5,258,668.**

[13] **FIG. 2** shows another prior-art embodiment of the cascading method.

[14] **FIG. 3** shows a first embodiment of the present invention.

[15] **FIG. 4** shows another embodiment of the present invention.

[16] **FIG. 5** shows cascading of the PLBs according to an embodiment of the invention.

[17] **FIG. 6** shows another example of PLB cascading as applied to a particular function according to an embodiment of the invention.

Detailed Description

[18] To avoid complications in the drawings and description, the invention is discussed with the simplest embodiments, but other more complicated embodiments will be evident to a person ordinarily skilled in the art. Therefore, the following description of the present invention is only illustrative and not in any way limiting.

[19] **FIG. 1** and **FIG. 2** have already been discussed in the background.

[20] **FIG. 3** shows one of the embodiments of the invention. The LUT ***210*** has four inputs **(A, B, C, D)** and produces the output LUTOUT ***251***. The output ***251*** of the LUT ***210*** is fed to the cascade logic ***220***. A multiplexer ***230*** is provided to connect either the cascade output ***253-O*** or the LUTOUT ***251*** to the flip-flop ***240***. The configuration bit ***P2*** is a selection bit for multiplexer ***230*** for determining whether the multiplexer connects LUTOUT ***251*** or CasIn ***253*** to output ***254***.

[21] The cascade logic element ***220*** has input ***253-I***, which connects to the cascade output ***253-O*** from another, preferably, adjacent cascade logic cell (not shown in **FIG. 3**). The cascade logic element ***220*** combines both the inputs LUTOUT and CasIn and provides a resulting output signal Casout ***253-O***.

[22] The output ***254*** of the multiplexer ***230*** is fed to the input of the flip-flop ***240***. The same line ***254*** is extended to one of the inputs of multiplexer ***250.*** Flip-flop ***240*** serves as the second input to multiplexer ***250***. Depending on the value of programmable bit ***P1***, multiplexer ***250*** selects either the signal on line ***254*** or the signal on line ***255*** as the logic module output on line ***256***. This output on line ***256*** is fed back to the LUT ***210*** by line ***257***.

[23] Multiplexer ***230*** provides flexibility for selecting either the direct output ***251*** of the LUT ***210*** or the cascade output **Casout *253-O***. This implementation can be used to obtain any sub function of the cascaded output Casout or direct output of the LUT ***210***, thereby eliminating the requirement of repeating or duplicating the same logic.

[24] **FIG. 4** shows another embodiment of the present invention. According to this embodiment, multiplexer ***320*** is provided at the cascade input ***343*** (referring to input line ***253-I*** of **FIG. 3**) before it is fed to the cascade logic cell ***330***. Multiplexer ***320*** is provided with one configurable bit ***P3*** as the select input. One input to the multiplexer is the cascade input **CasIn *345*** of the previous stage while the second input is its complement ***342.*** The remaining circuitry for cascading remains the same as described in **FIG. 3.** Although **FIG. 4** explicitly shows the cascade logic ***330*** as a

NAND gate, the logic ***330*** can be any desired logic gate or circuit. This flexibility is very useful to implement different type of complex functions, which may require that the cascade logic ***330*** implement functions other then AND or NAND.

[25] **FIG. 5** shows the schematic diagram of the cascade chain connectivity according to an embodiment of the invention. The cascade output **casout** ***445*** of the cascade block ***430*** of **PLB1** is connected to the **Casin** input of the **PLB2**. Similarly connectivity is repeated for the entire PLB array.

[26] **FIG. 6** shows an example of implementing a function **F = ~((A1+B1) * (A2+B2) * (A3+B3))** using cascade logic according to an embodiment of the invention. Three PLBs **PLB1, PLB2,** and **PLB3** are used for implementing this logic. The cascade logic ***530*** is configured for NAND operation. Therefore the LUT of **PLB1** is programmed to produce the inverted sum of the inputs $\mathbf{A_1}$ and $\mathbf{B_1}$ (a NOR operation). Similarly, The LUTs of **PLB2** and **PLB3** are programmed to produce the sum of the inputs $\mathbf{A_2}$**,** $\mathbf{B_2}$ and $\mathbf{A_3}$**,** $\mathbf{B_3}$ respectively (OR operations). The configuration bit **P** of multiplexer ***520*** of **PLB1** are configured as a "1" to pass initialization value "1", whereas the configuration bits **P** of multiplexers ***520*** of **PLB2** are configured to pass the direct cascade out of **PLB1** while **PLB3** is configured to pass the inverted version of the cascade out of the **PLB2** to produce function **F** at the cascade output ***545*** of the **PLB3**.

[27] One of the advantages of this architecture is that it allows simultaneous access to the LUT and cascade outputs. As shown in the example, the LUT outputs ***544*** of the respective **PLBs** as well as the cascade outputs ***545*** are available at the logic module output ***548*** of the respective programmable logic blocks, e.g. ~(A1+B1), A2+B2, or A3+B3 can be used as a sub-function to implement some other bigger functions.

[28] Furthermore, an Integrated Circuit (IC) such as a field-programmable gate array (FPGA) can incorporate the architectures of **FIGS. 3-6**, and an electronic system such as a computer system can incorporate the IC according to an embodiment of the invention.

[29] It will be apparent to those with ordinary skill in the art that the foregoing is merely illustrative intended to be exhaustive or limiting, having been presented by way of example only and that various modifications can be made within the scope of the above invention.

[30] Accordingly, this invention is not to be considered limited to the specific examples chosen for purposes of disclosure, but rather to cover all changes and modifications, which do not constitute departures from the permissible scope of the present invention. The invention is therefore not limited by the description contained herein or by the drawings.